論文
2-9-13

유닉스 클러스터시스템의 고속통신구조 상용화에 관한 연구 (High Speed Communication System for UNIX Cluster System)

김 현 철*
(Hyun-Chul Kim)

요 약

클러스터시스템의 표준 고속통신구조로서 Virtual Interface Architecture (VIA)가 일반적으로 제안되어진다. 그러나 현재 VIA 및 Virtual Interface Provider Library (VIPL)의 사양은 POSIX의 정해진 Fork 나 시그널 기능에 응답하는 규정이 없거나, 디스크립터가 잘못 규정되어지는 등, Windows OS와 Intel 아키텍처의 CPU에만 적합하도록 되어 있는 부분이 있다. 본 논문에서는 OS와 CPU 아키텍처의 중립적인 시각에서 VIA 및 VIPL의 문제점을 명확화 하고, 다른 OS나 CPU에의 상용화를 목적으로 하는 해결 방식을 제안한다.

ABSTRACT

The Virtual Interface Architecture (VIA) is usually suggested as a new standard high-performance communication of the cluster systems. However the VIA specification aims for platform independence, the current Intel VI Provider Library (VIPL) design favors systems with Intel architecture processors running the Windows operating system (OS). This paper aims for clarifying the guesstion problem of VIA and VIPL in the newtrul time of CPU Architecture and OS further more, It suggests the solution aiming the communication in the other CPU or OS

1. 서론

최근 고도의 안정성과 보수운용성이 요구되는 OLTP/OLAP 등을 위한 기술적 인프라 확산과 함께 인터넷 기술 분야의 폭발적 확산으로 관련 분야의 서비스를 제공하는 웹 기반의 서버 시스템들은 종래의 시스템과는 비교할 수 없을 정도의 고속 처리 능력과, 무정지 운영을 지향하고 있으며 이의 대표적인 기술로 고가용성인 보수운용성이 요구되어지고 있다.

이와 같이 고성능의 고가용성인 서버를 실현하는 방식으로서, 복수의 컴퓨터시스템을 고속결합으로 연결하여 하나의 서비스를 제공하는 클러스터시스템 연구 개발이 급속히 확산되어지고 있다.

클러스터시스템상에서 고성능의 병렬분산프로그램을 구현하는데는 복수의 노드상에 분산 배치되어지는 프로세스간의 통신 처리를 고속화 해야 하는 요구를 수반하며, 그 때문에 클러스터의 고속결합망으로서, System Area Network (SAN)이 개발 되어지고, SAN에 있어서의 고속통신 표준 사양으로서, Virtual Interface Architecture (VIA)가 제안되어진다.

* 정회원 : 대구산업정보대학 컴퓨터정보계열 겸임교원

논문접수 : 2001. 9. 15.
심사완료 : 2001. 9. 22.

그러나 VIA 및 Application Program Interface (API)에 있어 Virtual Interface Provider Library (VIPL)의 현재의 사양(Version 1.0)에는 특정 CPU 아키텍처와 OS에 대해서만 적용되는 규정을 내포하고 있어, 그 외의 다른 CPU나 OS상에서의 효율적인 VIA의 실현을 사실상 외면하고 있다. 본 논문에서는 OS나 CPU 아키텍처의 중립적인 시각에서 보는 현재의 VIA 및 VIPL 사양의 문제점을 명확화하고, 다른 OS나 CPU에 상용화 할 수 있는 해결방안을 제안한다. 그리고 실제로 UNIX 클러스터상의 고속통신구조를 실증함으로써 제안의 타당성을 확인한다. 본 논문에서 제안하는 개선안을 표준화 작업을 통해 업계 표준 사양에 반영되도록 하는데 그 목적이 있다.

2. SAN과 VIA

2.1 System Area Network (SAN)

종래의 클러스터시스템의 노드간 결합을 위한 네트워크로는 고속 LAN인 Fast Ethernet이나 FDDI 등이 주로 사용되어져 왔다. 하드웨어 측면에서는 LAN의 성능 향상이나 데이터 길이가 긴 통신에 있어서의 전송대역폭 확대 등 꾸준한 기술적 발전을 가져 왔으나 소프트웨어 적인 측면에서는 특히 프로토콜 부문에 있어서의 TCP/IP 등의 처리에서는 오버헤드는 삭감되었지만 성능 상으로는 그다지 큰 발전을 가져오지 못하고 있다. 특히 데이터 길이가 짧은 통신에서는 하드웨어를 고속화해도 통신시간은 단축되어지지 않고 있다.

소프트웨어적인 측면에서의 System Area Network (SAN)에는 메모리맵에 사용되어지는 제어 레지스터에 유저 프로세스로부터 직접 입력을 할 수 있는데, 이는 커널 모드에 천이 되지 않고 통신을 기동하는 것이 가능하기 때문이다.

또한 직접 다른 노드에 있는 주기억장치의 데이터를 읽고 쓰는 원격 DMA(Direct Memory Access) 기능을 갖추고 있으며, 프로그램의 사용 요청이 발생하면 통신 버퍼 간 데이터의 직접 전송(Zero-Copy)이 가능하다. SAN은 고속 LAN과 비교할 때 접속거리나 이 기종 접속성에는 상대적으로 성능이 낮지만 전송대역폭과 통신 시간에 관해서는 상대적으로 높은 성능과 낮은 에러 발생률을 갖추고 있다.

2.2 Virtual Interface Architecture(VIA)

VIA는 SAN의 하드웨어 자원을 직접 이용하여 통신을 하는데, 프로그램은 이 때 하드웨어를 점유하여 사용하기 좋도록 하는 일을 수행한다. VIA의 특징으로는 다수의 VI를 사용하는 프로그램을 복수개 동시지원 가능하다는 것이다. 각 VI는 송신-큐와 수신-큐를 기다리고, 통신 지시의 디스크립터를 큐에 만들어 연결하는 것으로 데이터 전송을 개시한다. 프로그램은 디스크립터의 어드레스에 쓰기 조작을 하기보다 커널을 경유케 하여 하드웨어에 조작을 의뢰한다.

VIA는 커넥션지향의 통신시스템을 제공하는데, 프로그램에서 VI로 불리는 통신 종단점을 만들고, 통신 전에 VI에 대하여 접속을 확립 시키는 통신을 수행한다. 이러한 병렬분산 프로그램에 있어서의 SAN의 이용 확대를 위해 고속통신구조의 표준사양으로서 Virtual Interface Architecture (VIA)가 제안되어진다.

3. VIA 사양의 중립화

VIA나 VIPL의 사양은 CPU 아키텍처 및 OS로부터의 독립성을 목적으로 설계되어져 있지만, 실제로는 해당 분야의 시장 점유율 등의 문제로 특정 CPU (Intel Architecture)와 OS(Windows NT)에서 구현되는 것에만 치중하고 있어 다른 CPU나 OS에서의 효율적인 구현을 위한 기술적 중립성에는 부족한 부분이 있다.

대규모 병렬 분산 프로그램은 각 프로그램 본체의 확장이나 보수를 용이하게 하기 때문에 본체부분으로부터 개개의 트랜스포트에 대응하는 통신처리부가 분리되어져 있는 경우가 많다.

새로운 트랜스포트에 있어 VIA의 이용은 기존의 소켓통신을 이용하고 있는 프로그램에 VIA용의 통신 처리부를 추가하여 실현한다.

VIA의 사양서가 하드웨어와 소프트웨어의 구조를 내포한 개요를 규정하는데 비해 VIPL의 사양서는 VIA사양에 기반한 구체적인 API 및 디스크립트 등의 주요한 데이터 구조의 메모리상의 포멧을 정의하는데 있다.

4. UNIX상에서의 구현 기술

4.1 메모리 등록

VIPL의 메모리 등록은 두 가지의 처리를 필요로 한다. 한 가지는 등록하는 메모리가 스와프가 되지 않도록 OS에 대해서 페이지 고정을 의뢰하는 것이고, 또 한 가지는 데이터통신을 제어하는 어드레스 변환을 보호할 필요가 있는 정보를 기록하는 것이다.

(1) 메모리 페이지의 고정

메모리를 고정한 상태에서 제어가 호출하는 조건에서 Solaris의 비동기 I/O용의 메모리 고정을 구현한다. 그러나 이런 방법으로 메모리를 고정한 프로그램은 예외사항이 발생해도 프로세스를 종료할 수 없기 때문에 프로세스 종료 신호를 검출하기 위해 전용 슬레트를 라이브러리 내에 이용하게 된다.

(2) 어드레스 변환

어드레스 변환과 보호에 필요한 정보는 장치 드라이버 내부에 제어 데이터로서 보관하고 있다. VI 하드웨어 구현에 있어서 VI 하드웨어는 무리적 어드레스를 사용하여 메인메모리에 접근한다. 그러나 SPARC를 사용하는 서버에서의 I/O 장치는 물리적 어드레스는 없고 DMA용의 가상 어드레스(Direct Virtual Memory Address, DVMA)를 사용하는 접근 방법이 필요하다.

DVMA는 복수 페이지에 걸쳐 선형에 분할되어진다는 이점을 가지고, 동시에 분할되어 사용되는 자원량이 제한되어 있다. Solaris 2.6에서는 PCI 버스 같은 곳에서는 최대 수십MB가 되고, 이런 이유로 프로그램이 메모리를 등록하는 영역 전체에 DVMA를 분할할 수 있다.

4.2 Zero-Copy 통신

(1) SEND-GET 통신

Synfinity-0의 원격 DMA 명령을 이용해 Zero-Copy 통신을 구현할 수 있는데, VIA의 원격 DMA형 통신만이 아닌 송수신형 통신도 Zero-Copy로 구현할 수 있다. 페이로드의 전송에 선행하여 DVMA의 분할로 수신측에 SEND명령을 사용해서 통신 의뢰를 송부하고, 수신측에서는 수신버퍼에 대해 DVMA를 분할케 한 후 GET 명령을 이용하여 페이로드 전송을 한다.

Zero-Copy 통신 방식을 SEND-GET 통신이라고도 하는데, SEND-GET 통신에서는 CPU에 페이로드 복사 처리가 필요하지 않는 반면 양방향 노드에 DVMA의 분할과 파라미터를 수용한 SEND통신이 필요하다. 본 연구에서는 페이로드 길이에 비례하지 않은 고정값을 가지고서 페이로드 길이의 짧은 전송에는 적용하지 않는다. 또한 GET 명령은 페이로드의 통신에는 사용하지 않는다.

(2) COPY-SEND-COPY 통신

페이로드의 길이가 짧거나, 얼라이먼트가 일치하지 않는 곳에는 커널 내의 버퍼를 경유하는 통신방식을 사용하게 되는데, 우선 송신측에는 프로그램의 송신 버퍼로부터 커널 내의 송신 버퍼에 페이로드 복사 제어헤드를 부가한다. 그리고 SEND 명령보다 제어헤드와 페이로드를 수신측의 커널 내에 있는 수신 버퍼에 전송한다. 수신측에는 페이로드를 커널 내의 버퍼로부터 프로그램의 수신 버퍼에 재차 복사한다. 이런 통신방식을 COPY-SEND-COPY 통신이라고 부른다.

커널 내 버퍼와 초기화시에 DVMA를 나누어 주는 것으로, 전후의 복사처리 보다 얼라인먼트를 조정하는 것으로, 얼라인먼트와 일치하지 않는 페이로드를 전송할 수 있다.

5. 통신 성능 평가

<표 1>의 환경을 이용해서, 기본적인 통신 성능으로서 편방향 통신에 필요한 시간(대기시간)과 데이터 전송 대역폭을 측정하고, TCP/IC 통신과 성능 비교를 실행한다. 또한 4장에서 설명한 2종류의 수송방식의 상세한 처리시간분석 및 사용자 수준 통신의 효과를 측정한다.

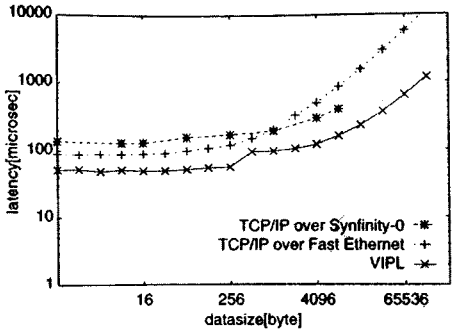
<표 1> 성능 측정 환경

<Table 1> Measurement system specifications

CPU	UltraSPARC-II 360 MHz
CPU 수	2개 / Node
SAN	Synfinity-0
Memory 수	512MB / Node
Node 수	2

(1) 대기시간

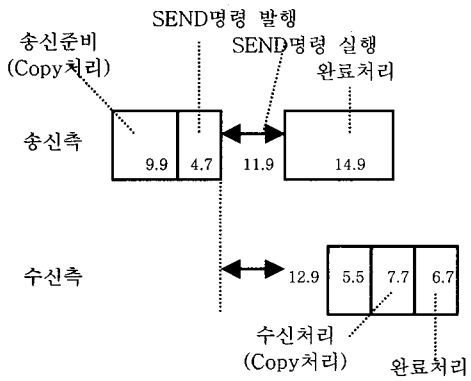
Synfinity-0 상에서 작성한 VIPL과 TCP/IP의 프로그램으로부터 나타난 편방향 통신시간(대기시간)을 페이로드 길이가 변해져 측정한 결과를 [그림 1]에서 알 수 있다. TCP/IP의 측정에는 통신시간 측정 프로그램을 사용하고, Fast Ethernet 및 Synfinity-0 상의 측정을 진행한다. VIPL의 측정에는 UCB의 LogP 모델에 기본하여 작성한 프로그램을 사용한다.



[그림 1] VIPL과 TCP/IP의 대기시간 성능비교

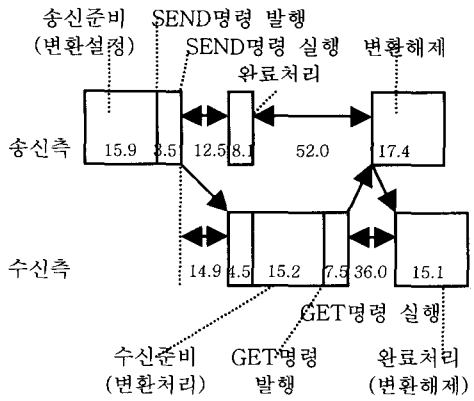
[Fig. 1] Latency comparison.

페이로드 길이가 긴 전송에는 물론 512Byte 이하의 짧은 전송에도, VIPL은 TCP/IP에 비해 대기시간이 약 반정도 단축되어진다. 1Byte 전송 시간은, Fast Ethernet 상의 TCP/IP가 88μ초 정도의 응답이 나타나고, VIA는 51μ초이다. 더욱이 디스크립터를 사용한 0Byte의 통신은 42μ초 밖에 걸리지 않는다. 4장에서 설명한 COPY-SEND-COPY 통신과 SEND-GET 통신과 관련한 제어 페이로드 헤드도 512Byte에 설정하고 있다. 두 번째 전송방식에의 각 처리에 걸리는 CPU 시간의 내역은 [그림 2]와 [그림 3]에서 알 수 있다. 각 처리의 개시 및 종료부분에 대한 시각을 조사한다. 그 차이를 계산하여 각 처리에 걸리는 시간을 구한다. 이 때 실험은 복수로 처리하여 그림에서는 그 평균치를 나타내고 있다. 사각형 표시가 실제로 디바이스 드라이버가 실행한 처리를 표시하고 있다. 측정이 불가능한 시각 취득의 오버헤드는 처리시간 전체의 5% 이하이다.



[그림 2] COPY-SEND-COPY 통신(256Byte)의 처리 내역(μ초)

[Fig. 2] Time slice of COPY-SEND-COPY protocol



[그림 3] SEND-GET 통신(4KB)의 처리 내역(μ초)

[Fig. 3] Time slice of SEND-GET protocol

[그림 2]는 256Byte 의 페이로드를 전송할 때의 COPY-SEND-COPY 통신의 송신측 및 수신측에 실행하는 처리시간을 보여주고 있다. [그림 2]에 나타난 결과는 아래와 같다.

- 송신측에는 (a) 커널 버퍼의 데이터 복사 시간을 포함한 송신준비, (b) SEND 명령의 발행처리, (c) SEND 명령의 실행, (d) 완료처리로 합계 41.4 μ초가 걸린다.
- 전달시간과 Solaris내부의 처리에 12.9 μ초가 걸린다.
- 수신측에는 (b) 프로그램 버퍼에 데이터 복사 시간을 포함한 수신처리, (c) 완료처리에서 합계 19.9 μ초가 걸린다.

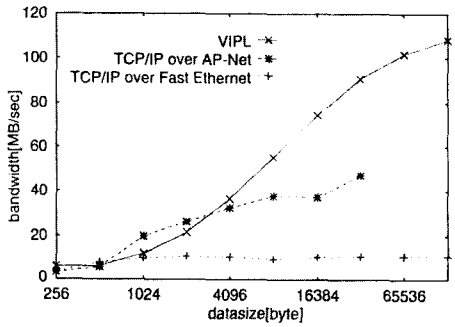
[그림 3]은 4KB의 페이로드를 전송하의 SEND-GET 통신에서의 처리시간을 나타내고 있다. [그림 3]에 나타난 결과는 아래와 같다.

- 송신측에는 (a) DVMA의 할부 시간을 포함한 송신준비, (b) SEND 명령의 발행처리, (c) SEND 명령의 실행, (d) 완료처리로, 합계 40.4 μ초가 걸린다.
- SEND 명령의 전달시간과 Solaris 내부 처리에 14.9 μ초가 걸린다.
- 수신측에는 (a) DVMA 할부시간을 포함한 수신준비, (b) GET 명령의 처리 완료, (c) GET 명령의 처리시간, (d) DVMA의 해제시간을 포함한 완료처리로써 합계 78.3 μ초가 걸린다.
- GET 명령의 종료를 계기로, 송신측에도 DVMA의 해제를 포함한 완료처리에 17.4 μ초가 걸린다.

[그림 3]에서 보여지는 처리시간 중에서, DVMA의 설정과 해제의 처리시간은 각기 7.2 μ초와 3.6μ초 인데 1회의 편방향 통신에 맞는 합계로 18.0 μ초 (13.5%)의 오버헤드가 있는 것을 알 수 있다. Solaris7 부터는 Dual Address Cycle을 지원하는 하드웨어에 DVMA에 대해 물리 어드레스를 사용할 수 있는데 이것으로 DVMA처리를 생략해서 통신을 보다 고속화 할 수 있게 된다.

(2) 전송대역폭

Synfinity-0 상에서 작성된 VIPL과 TCP/IP의 데이터 전송폭을 페이로드 길이를 변화해서 측정한 결과를 [그림 4]에서 알 수 있다. 동일한 Synfinity-0를 사용해도 페이로드 길이 32KB의 전송에서 TCP/IP의 47.3 MB/S에 비해 1.9배의 90.6MB/S의 성능을 나타내고 있다.



[그림 4] VIPL과 TCP/IP의 데이터 전송 대역폭 성능 비교

[Fig. 10 Bandwidth comparison

(3) 사용자수준의 통신

<표 2>를 통해 사용자 수준 통신의 효과를 알 수 있다. 동일한 송신조작(및 수신 조작)에 있어서도, 시스템 콜 발행을 생략할 수 있는 것은 실제의 시스

템콜을 발행하는데 있어서 커널내에서 처리하는데 비해 처리완료까지의 시간이 대폭 삭감되어지는 것을 알 수 있다.

<표 2> 사용자 수준 통신의 효과

<Table 2> User level transfer benefit

System Copy	실행 시	생략 시
송신처리 (Vip Post Send)	29.1 μ초	1.5 μ초
수신처리 (Vip Post Recv)	10.1 μ초	1.2 μ초

7. 결론

본 논문에서는 고성능통신시스템의 표준으로서 제안되어지는 VIA 및 VIPL의 현재의 사양에 대해서, CPU 아키텍처 및 OS로 부터의 중립성에 부족한 부분을 명확화 하였다.

실제로, SPARC Solaris상에 Synfinity-0를 이용하여 고성능의 VIA를 실증하고, 복수의 대규모 상용병렬 프로그램으로 실제로 VIA를 사용해서 기능 확장 제안의 유효성을 확인하였다. 또한 VIA의 기본통신 성능의 측정과 해석을 시행하여 양호한 성능을 얻을 수 있음을 보였다.

향후, 실용 병렬 프로그램이 VIA에서 어디까지 고속화할 수 있는가를 평가하여야 할 것이며, 현재 장치 드라이버에 에뮬레이션 처리를, 전부 하드웨어에서 직접 실행하는 SAN이 개발되어야 할 것이다.

※ 참고문헌

[1] Intel Corp. "Intel Virtual Interface Architecture Developer's Guide Revision 1.0 (1998)

[2] Saletore, V., et al. "Introducing VI Develoeper Forum, 1999 Fall Intel Developer Forum (1999)

[3] Dunning, A., et al. "The Virtual Interface Architecture, IEEE Micro, Vol.18, No.2, pp.66-77 (1998)

[4] National Energy Research Scientific Computing Center "M-VIA "A High Performance Modual VIA for Linux Release Notes, Berkeley CA (1999)

[5] Compaq Computer Corp., Intel Corp. and Microsoft Corp. "Virtual Interface Architecture Specification version 1.0 (1997)

[6] Culler, D., et al. "Assessing Fast Network Interfaces, IEEE Micro, Vol.16, No.1, pp.35-43 (1996)

[7] Brown, G. "Lessons from a VIA 0.9 Implementation, Proc. Hot Interconnects V, IEEE CS (1997)

[8] 김성환 "UNIX System Contact", pp677-689, 도서출판 대림 (2000).

김 현 칠

1994 경일대학교 전자계산학과 공학사

1998 영남대학교 경영대학원 경영학과 경영학석사

2001 영남대학교 대학원 경영학과 박사과정

1999 ~ 현재 대구산업정보대학 컴퓨터정보계열 겸임교원